

# 8-Bit Arithmetic Logic Unit

Jonathan Gerrans, Department of Electrical and Computer Engineering  
University of Maine, Orono

## CONTENTS

<b>Abstract</b>		1
<b>I Introduction</b>		2
A Project Overview . . . . .		2
B Objectives . . . . .		2
<b>II Circuit Design</b>		2
A Logic Gates . . . . .		2
B 4-Bit ALU Logic Circuit . . . . .		3
C Verification . . . . .		4
<b>III Layout</b>		4
A ALU Layout . . . . .		4
B ALU With Output Buffers . . . . .		5
C Chip Layout . . . . .		6
<b>IV Chip Verification</b>		6
<b>V Physical Chip</b>		7
<b>VI Chip Testing</b>		8
A Setup . . . . .		8
B Count Test . . . . .		9
C Arithmetic and Logic Tests. . . . .		10
<b>VII Future Improvements</b>		10
<b>VIII Conclusion</b>		10
<b>References</b>		11
<b>Bio</b>		11

### Abstract

The design, layout, and testing of an 8-bit Arithmetic Logic Unit (ALU) based on the 74LS181 4-bit ALU circuit from Texas Instruments is discussed. Two 4-bit ALUs were cascaded together to create one 8-bit ALU on-chip. Details of the design process are included from each step in the project, along with diagrams, photos, and renders showing the various components and testing setups. The project was undertaken as a means to learning the VLSI design process, and resulted in a fully functional ALU with the same functionality as a commercially available product. Minimal problems were encountered with the final design, and are addressed.

## I. INTRODUCTION

### A. Project Overview

This ECE 547 VLSI Design project is an 8-bit ALU designed using a 180nm process in Cadence's Virtuoso platform. The circuit was based on Texas Instruments' 74LS181 [1] 4-bit ALU design. Two copies of the circuit were placed on the chip and cascaded to create an 8-bit ALU. The inputs to the ALU are two 8-bit input busses, A and B, as well as a 4-bit function select bus, mode select, carry in, and clock. The outputs include the 8-bit F output bus, and carry out, carry propagate, carry generate, and A=B. The ALU features 16 arithmetic and 16 logic operations, in both positive and negative logic, and has support for high-speed carry lookahead operation when used in conjunction with the 9342 or 93S42 circuit. There was no specific goal for operational frequency, however, in simulation, speeds of up to 500 MHz were achieved, and in testing, the circuit performed well up to the maximum 80 MHz allowed by the function generator. The fabrication was handled through MOSIS, using IBM's CMRF7SF process, and the device was packaged in a 40-pin ceramic DIP.

### B. Objectives

The main objective throughout this project was to become familiar with VLSI design. Thus, there were no high goals set for this project to attain. Simply designing a fully functioning 8-bit ALU was considered sufficient for the scope of this course. The goal was to implement the full functionality of the 74LS181 chip and test it at various speeds to determine the efficiency of the unit.

## II. CIRCUIT DESIGN

A number of different logic gates were custom designed for this chip. As the schematic is quite complex, gates with multiple numbers of inputs were needed. The different logic gates used in the design will be discussed first, followed by the schematic for the 4-bit ALU.

### A. Logic Gates

CMOS technology was used to design all the circuit elements of this chip. For most of the gates, the minimum sizing was used in the design, but several larger gates were needed where more than 3-4 elements were driven by the gate. An extra wide inverter was needed, as well as a 2x width NOR and a 3x width NOR, for various places in the circuit. Following is the list of gates (and quantities) used in the design of the 4-bit ALU:

- Inverter (47 -- 38 were placed on the outputs of NANDs to create ANDs)
- 1-input NANDs (5)
- 2-input NANDs (17)
- 3-input NANDs (12)
- 4-input NANDs (5)
- 5-input NANDs (2)
- 2-input NORs (6)
- 3-input NORs (5)
- 4-input NORs (2)
- 2-input XORs (4)
- T-Flip-Flops (8)
- Output Buffers (7)

Once all the logic gates and other components were designed, work started on assembling all the units together into the full ALU circuit. As mentioned previously, the design was taken from the Texas Instruments 74LS181 circuit. The circuit in its entirety was reproduced in the schematic, in order to gain the most experience with the process.



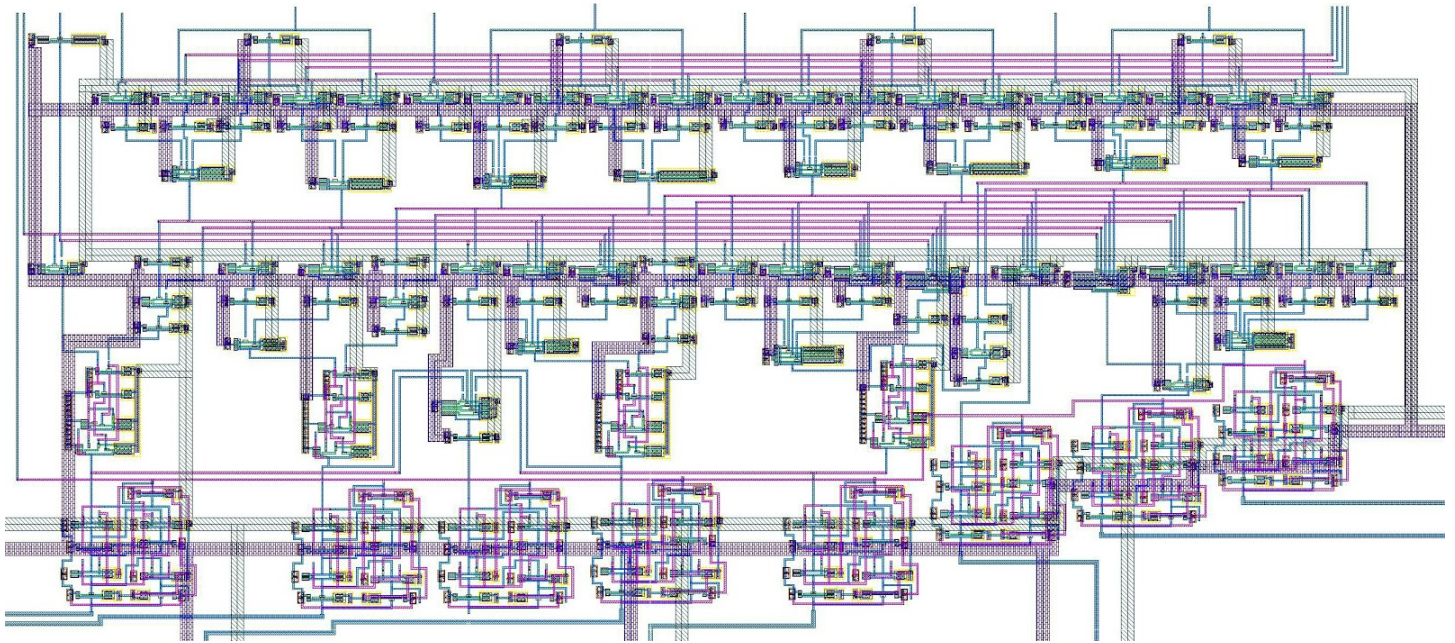
### C. Verification

During each step of the process, every circuit element was tested. Each logic gate, each step in the assembly process, and the full 4-bit ALU, were tested to assure the proper functionality. All the testing was done manually, so exhaustive testing of the final circuit was not done, but a few functions were verified before it was sent for fabrication. Several errors were discovered and remedied, and once the design worked in all test cases, it was submitted.

## III LAYOUT

### A. ALU Layout

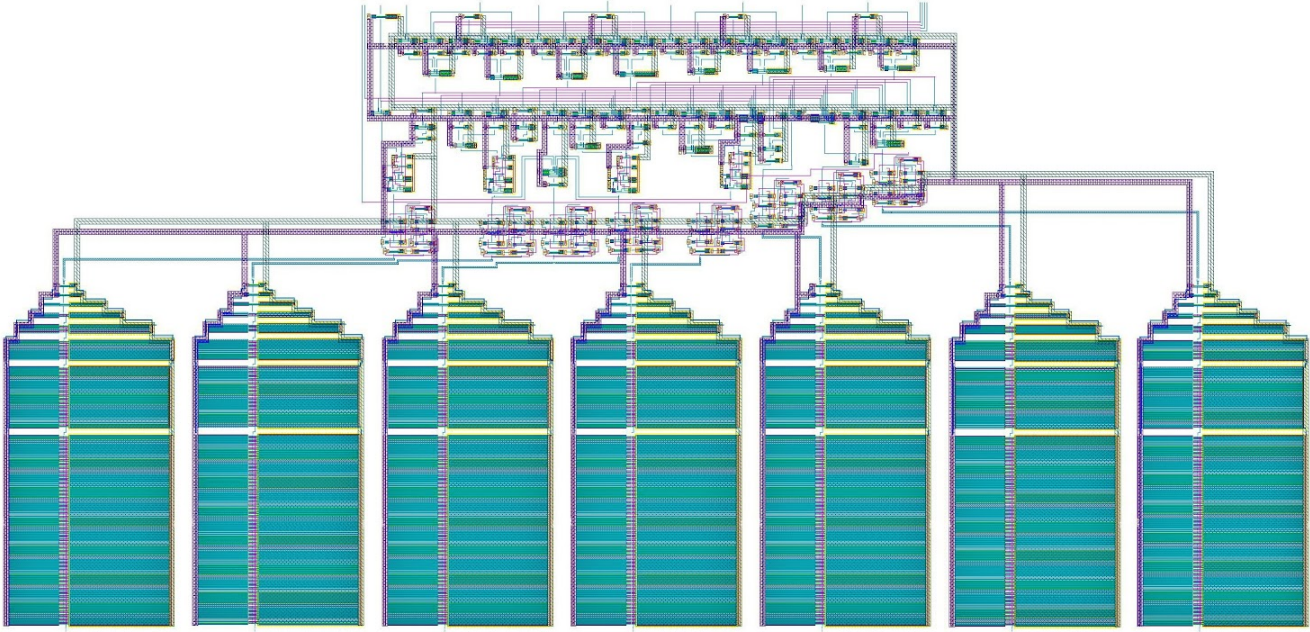
IBM's CMRF7SF design process includes 6 metal layers. The design of the ALU was limited to Metal 1 and Metal 2, to allow for higher level circuitry overlay if needed. Metal 1 was used for east-west wiring, and Metal 2 was used for north-south wiring, in a bus-style architecture. The ALU layout can be seen below:



This image is displayed on its side. The circuit was designed vertically, with the busses to the left and the t-flip-flops to the right. In this render, the two bussed lines of gates are the two levels of the ALU. The four larger gates below are the XORs, and the eight clusters across the bottom are the t-flip-flops. The layout was done to match the spatial arrangement of the logic diagram, to reduce confusion. A more space-efficient design could be used, but at the expense of much more time spent.

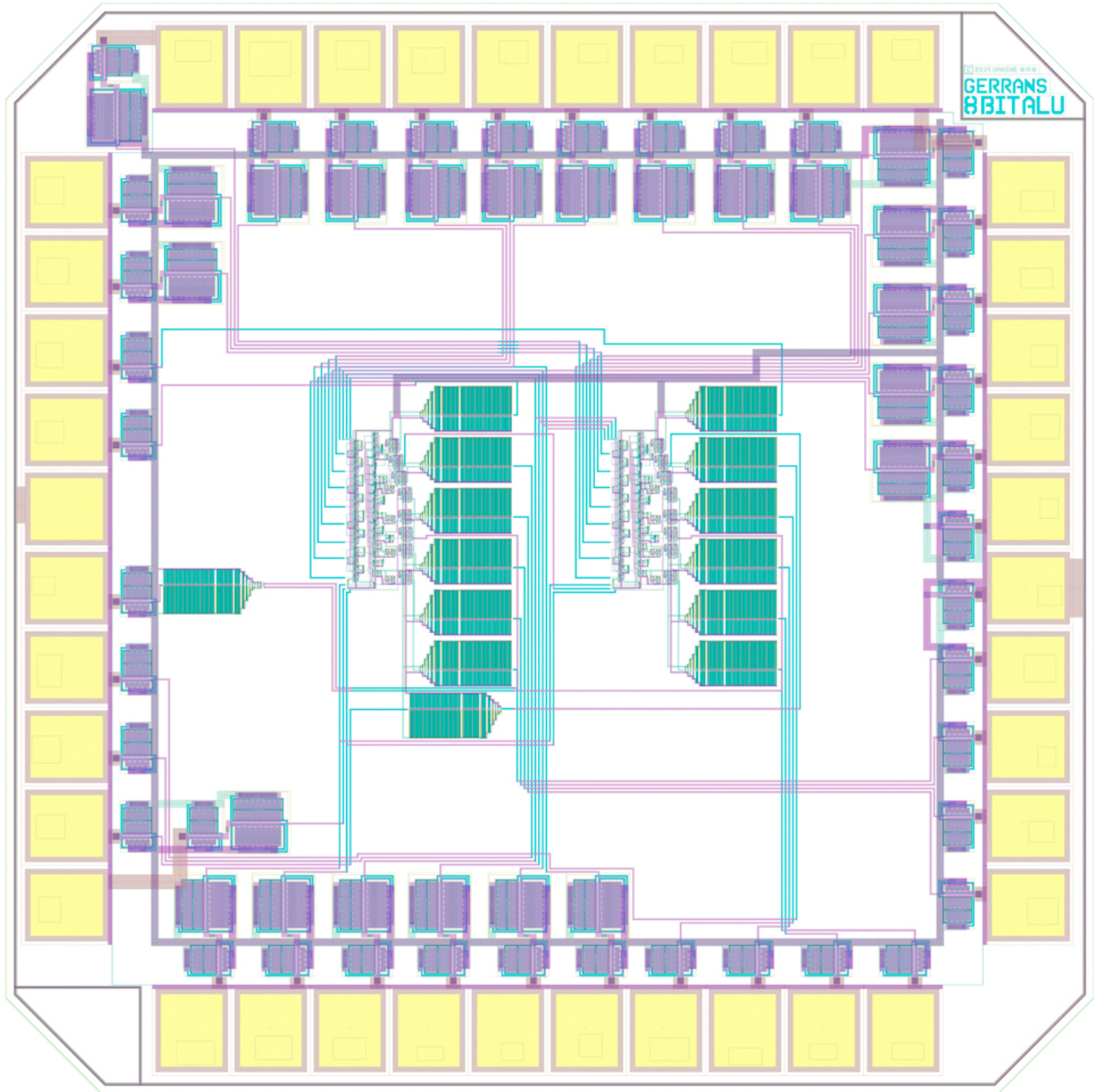
### B. ALU With Output Buffers

Below is a render with the output buffers included. Buffers are required to ramp the output signals up enough to be able to drive the pins on the package. An eight-stage buffer was required in this case. Note the relative size of the buffers to the ALU circuit, even though the circuit could be even more compact. The buffers occupied a large percentage of the total area of the ALU.



### C. Chip Layout

Below is the full chip layout. Metal 4 and 5 were used for power and ground, and interconnects were implemented with metal 1 and 2, due to the amount of space still free on the chip. This made visualizing the circuit easier as the layout progressed. The two ALUs can be seen in the center of the chip, with the left-hand one being the first stage and the right-hand one being the second stage, with the carry out of ALU1 connected to the carry in of ALU2. Two more buffers were added in addition to the ones in the ALU design; one was for the carry output, and the other was for the A=B output, which was the result of the A=B lines from each ALU combined together. The structures around the inside edges of the pads are ESD protection circuitry elements. Nearly all of the pins were used for the design, with 39 out of 40 pins occupied.

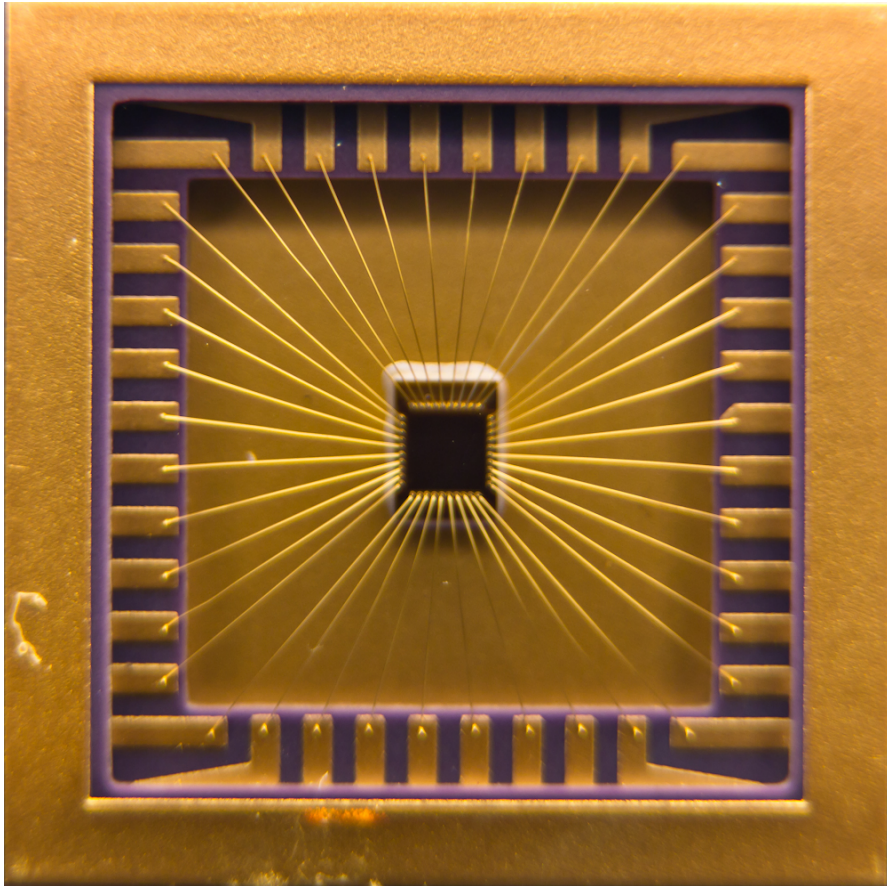


## IV CHIP VERIFICATION

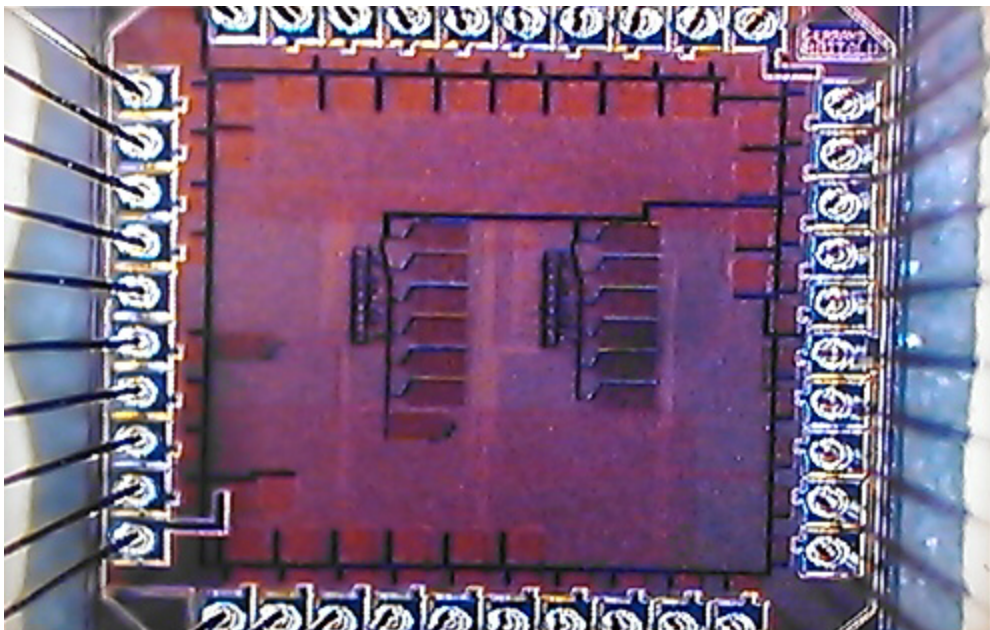
Further testing was done on the full chip design, but due to an error in the software, the full layout could not be tested with extracted parasitics. The chip was tested without extracted parasitics, by implementing a counter using the A+B operation with A set to 1 and B set to the F outputs, and it ran up to frequencies of 500 MHz before the waveforms started to distort.

## V PHYSICAL CHIP

The chip from this project as well as other students' projects were sent to MOSIS over the summer to be fabricated. In the fall, the chips were mailed out to each student. About half were mounted in a ceramic 40-pin DIP, and the remainder were sent unpackaged. Below are some images of the fabricated chips.



The chip mounted inside the socket in the 40-pin DIP

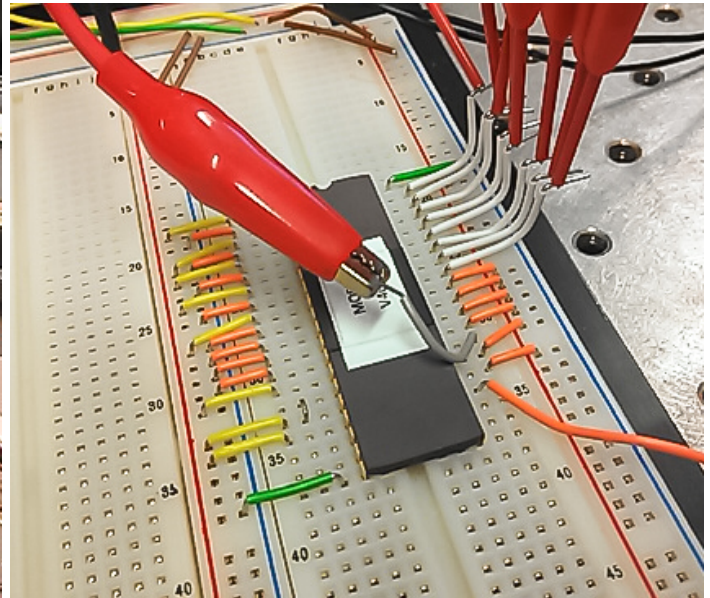
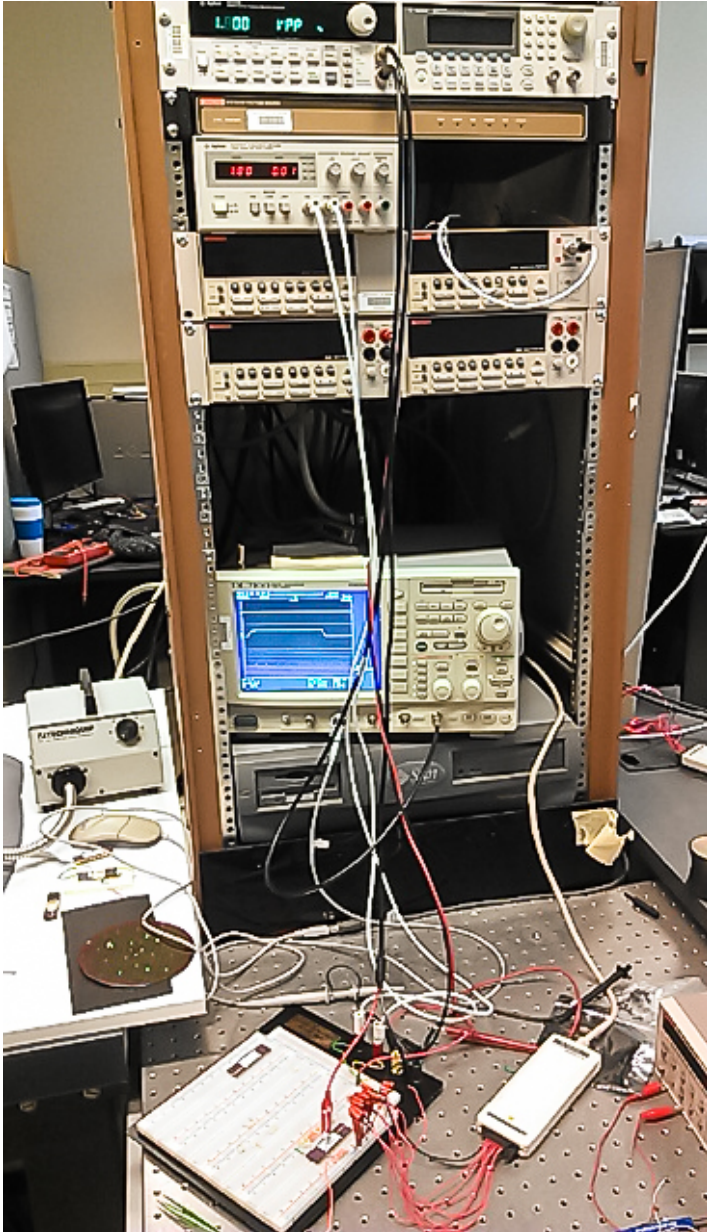


An image of the chip taken using a microscope. The buffers can easily be seen, as well as one of the power metal layers.

## VI CHIP TESTING

### A. Setup

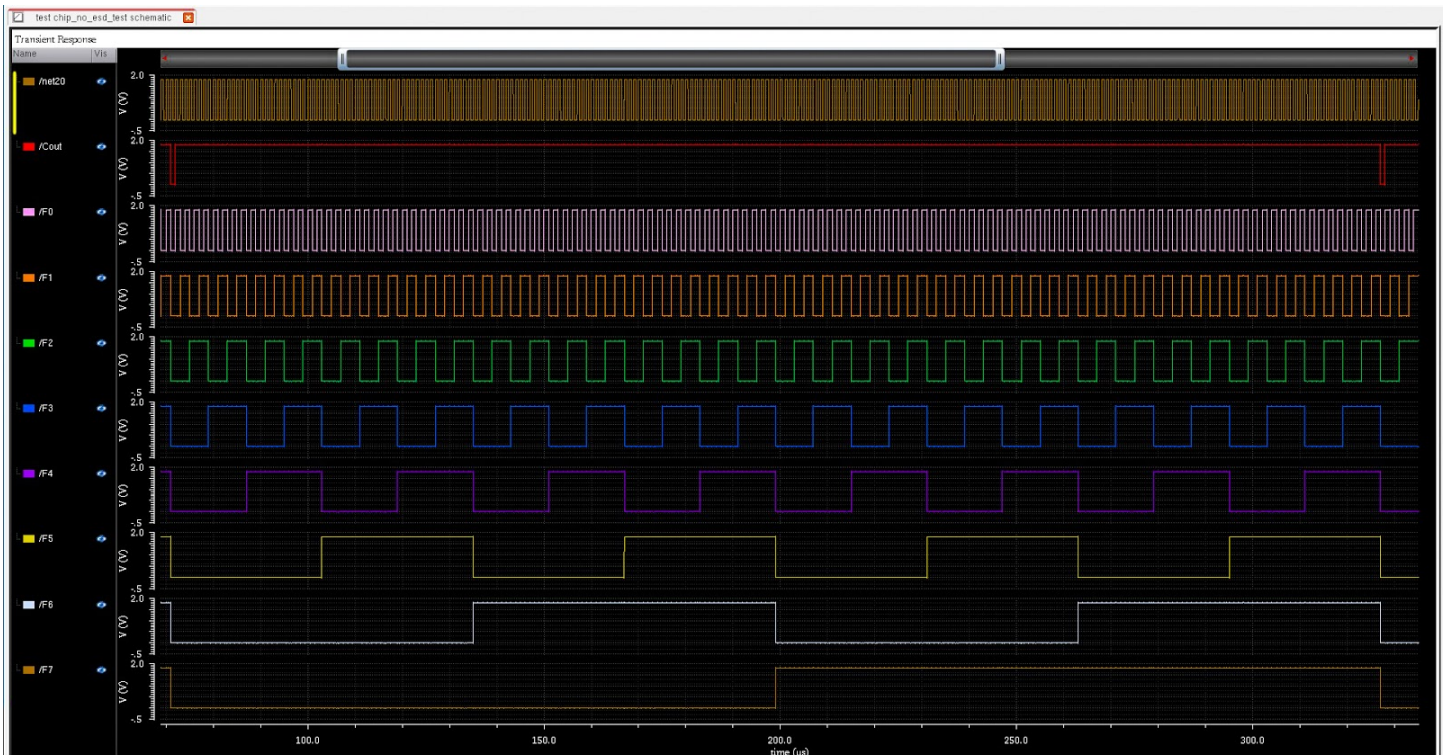
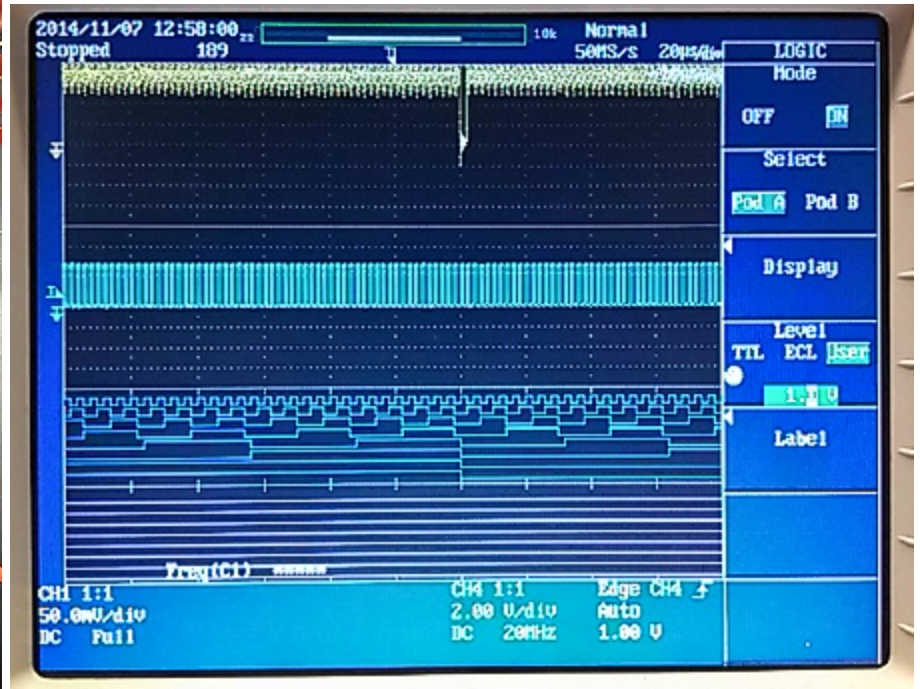
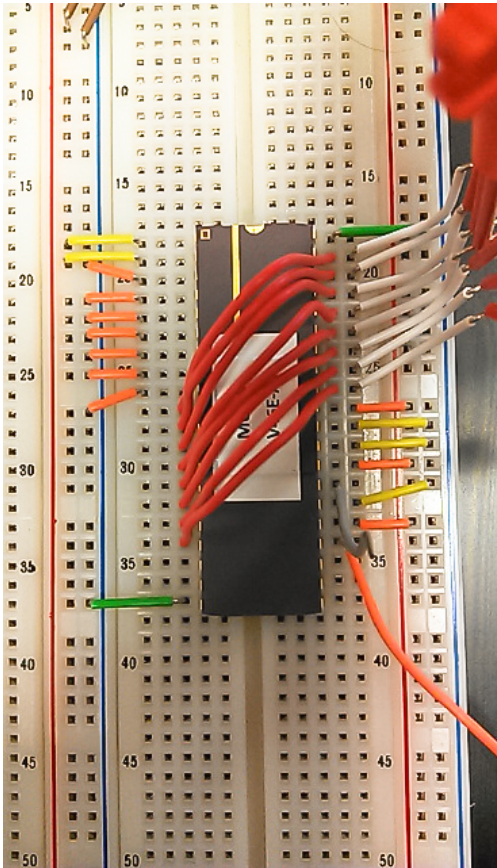
Below is an image of the setup used for testing the chip. A packaged chip was used for testing, and the instruments used included a power supply, a function generator, and an oscilloscope. The function generator provided the clock signal. An 8-bit logic wire harness was used to read the outputs from the chip and feed them into the oscilloscope. A breadboard was used to wire up the chip. Red rails are power and blue rails are ground, allowing different logic to be inputted to the chip by switching out wires.



Photos of setup, showing instruments and wiring for chip

### B. Count Test

The first test implemented on the chip was a count test. The A input was set to logic 1, and the F outputs were wired to the B inputs. A+B was selected with the select lines, and thus a simple counter was implemented. The clock speed was originally set to 1 MHz. The chip drew  $\sim 0.01A$  current at that frequency. The frequency was increased to the maximum of 80 MHz and the current draw increased to 0.02A around 80 MHz. The below photos show the wiring and oscilloscope results, compared to a software simulation.



During the count test, a one-cycle delay was observed between the low 4 bits and the high 4 bits. It was determined that a t-flip-flop was included between the carry out of the low ALU and the carry in of the high ALU, which was causing the delay. In software, the flip-flop was removed, and the simulation render shows the result, with no delay between the two halves of the full ALU.

### C. Arithmetic and Logic Tests

After the count test was run, a few other arithmetic operations were tested with positive results. Then a thorough testing of the positive logic operations was conducted. Each of the 16 logic operations was tested, with photo documentation of the wiring as well as the oscilloscope reading. The photos are excluded for brevity, and a chart showing the test results is below:

ACTIVE HIGH LOGIC						
Function	S3..S0	A	B	Expected F	Simulated F	Actual F
$\sim A$	LLLL	10101010	11110000	01010101	01010101	10101010
$\sim(A + B)$	LLH	10101010	11110000	00000101	00000101	00000101
$\sim AB$	LLHL	10101010	11110000	01010000	01010000	01010000
Logic 0	LLHH	10101010	11110000	00000000	00000000	00000000
$\sim(AB)$	LHLL	10101010	11110000	01011111	01011111	01011111
$\sim B$	LHLH	10101010	11110000	00001111	00001111	00001111
A xor B	LHHL	10101010	11110000	01011010	01011010	01011010
$A \sim B$	LHHH	10101010	11110000	00001010	00001010	00001010
$\sim A + B$	HLLL	10101010	11110000	11110101	11110101	11110101
$\sim(A \text{ xor } B)$	HLLH	10101010	11110000	10100101	10100101	10100101
B	HLHL	10101010	11110000	11110000	11110000	11110000
AB	HLHH	10101010	11110000	10100000	10100000	10100000
Logic 1	HHLL	10101010	11110000	11111111	11111111	11111111
$A + \sim B$	HHLH	10101010	11110000	10101111	10101111	10101111
$A + B$	HHHL	10101010	11110000	11111010	11111010	11111010
A	HHHH	10101010	11110000	10101010	10101010	10101010

## VII FUTURE IMPROVEMENTS

The operation of the 8-bit ALU was determined to be nearly 100% satisfactory. There was only one error detected during testing, which was the aforementioned flip-flop on the internal carry line. That has been removed in the design, and if an updated chip were to be fabricated, it is clear that the problem will have been solved. One other modification that could be made, which would aid in debugging and testing of the ALU, would be adding the internal carry line to the output pins. There is one remaining unused pin on the package, and if it were tied to the carry line between the two ALUs it would have aided in determining the cause of the delay that was mentioned. Furthermore, during complex operations it would be useful to have the ability to observe the behavior of that line.

## VIII CONCLUSION

In conclusion, the project as a whole was a success, with only one error discovered in the design of the chip, one which is not detrimental to much of the operation of the chip, and could be accounted for in design. Clearly if an updated chip were to be fabricated, the problem could be amended completely, but this is cost-ineffective, and will likely not be done. The goals of the project were met, in that a working ALU was designed from circuit diagram to working chip, and the basics of VLSI design were experienced.

## REFERENCES

- [1] SN74LS181 Arithmetic Logic Units/Function Generators Datasheet, Dallas, Texas, 1988, Available: <http://www.ti.com.cn/cn/lit/ds/symlink/sn74ls181.pdf>

**Jonathan Gerrans** received a B.S. in Computer Science from Southern Adventist University in Collegedale, TN in 2010. He is currently completing the last requirements of his M.S. degree in Computer Engineering at the University of Maine, Orono. His interests are computer hardware design, specifically PCB and chip design. He is originally from Brunswick, Maine.